

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-202155

(43)Date of publication of application : 27.07.2001

(51)Int.Cl.

G06F 1/06

G06F 1/04

(21)Application number : 2000-009303

(71)Applicant : HITACHI LTD

(22)Date of filing : 18.01.2000

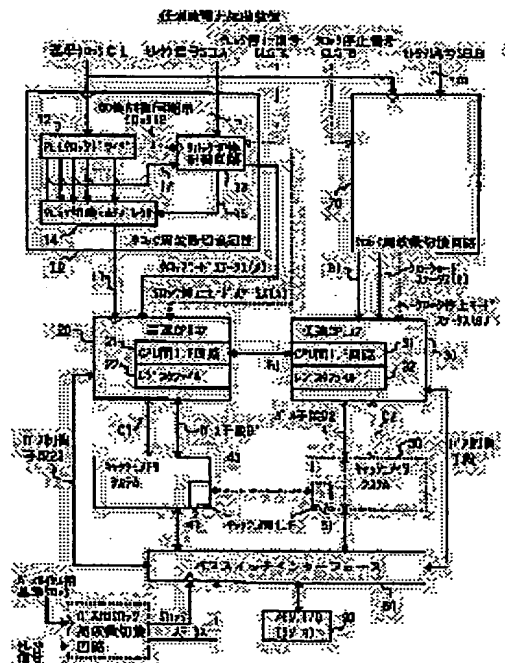
(72)Inventor : KAMETANI MASATSUGU
 UMEKITA KAZUHIRO
 YAMAMOTO KENJIRO
 KOYAMA MASAHIRO
 MOMOI YASUYUKI
 FUNATSU TERUNOBU
 IKE KATSUHISA

(54) LOW POWER CONSUMPTION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To finely control power consumption of a processor by enabling high speed, smooth and dynamic switching of multiple clocks and further enabling high speed and dynamic switching of multiple CPUs.

SOLUTION: High speed and smooth switching of clocks is realized by generating a switching signal 15 the clock levels of which before and after switching are the same and which is synchronized with a standard clock CL by a clock switching control circuit 13, taking out a clock from a PLL clock driver 12 by selecting it by a multiplexer 14 by the switching signal 15. The switching is dynamically executed by outputting select signals SELA, SELB by execution of an instruction of the CPU. In the switching between CPUs 20, 30, when the CPU under operation executes a switching instruction of the CPUs, supply of clocks to other CPUs is started, the present CPU enters a stopped state and the CPUs to which the clocks are supplied stop the clocks of other CPUs and starts operation of the present CPU.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the
 examiner's decision of rejection or application converted
 registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of
 rejection]

[Date of requesting appeal against examiner's decision of
 rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-202155

(P2001-202155A)

(43) 公開日 平成13年7月27日 (2001.7.27)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード(参考)
G 0 6 F 1/06		G 0 6 F 1/04	3 0 1 C 5 B 0 7 9
1/04	3 0 1		3 1 0 A

審査請求 未請求 請求項の数 7 O L (全 15 頁)

(21) 出願番号 特願2000-9303(P2000-9303)

(22) 出願日 平成12年1月18日 (2000.1.18)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 亀谷 雅嗣

茨城県土浦市神立町502番地 株式会社日

立製作所機械研究所内

(72) 発明者 梅北 和弘

茨城県土浦市神立町502番地 株式会社日

立製作所機械研究所内

(74) 代理人 100093872

弁理士 高崎 芳紘

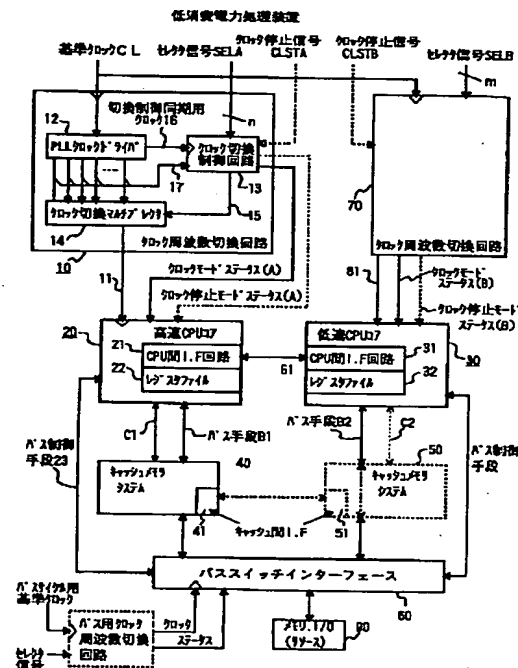
最終頁に続く

(54) 【発明の名称】 低消費電力処理装置

(57) 【要約】

【課題】 複数クロックを高速かつスムーズに動的に切換え可能とし、さらに複数のCPUの高速で動的切換え可能とし、処理装置の電力消費量をきめ細かく制御する。

【解決手段】 クロック切換制御回路13により切換前後のクロックレベルが同一でかつ基準クロックCLに同期した切換信号15を生成し、これによってPLLクロックドライバ12からのクロックをマルチプレクサ14で選択してとり出すことにより、高速かつスムーズなクロック切換を実現する。この切換はCPUの命令実行によりセレクト信号SELA、SELBを出力することでダイナミックに実行する。CPU20、30の切り換えは動作中CPUがCPUの切換命令を実行すると相手CPUのクロック供給をスタートさせて自CPUは休止状態となり、クロック供給されたCPUは相手CPUのクロックを止めて自CPUの動作を開始する。



1

【特許請求の範囲】

【請求項1】 与えられた基準クロックに同期しかつ互いに異なる周波数を持つ複数のクロックを生成するPLLクロックドライバと、

このPLLクロックドライバにより生成されたクロックの内の1つを入力された切換信号に応じて選択し、切換後クロックとして出力するクロック切換マルチプレクサと、

このクロック切換マルチプレクサにより切り換えの前に選択されていた切換前クロックと前記切換後クロックが同じ信号レベルにありかつ前記基準クロックに同期したタイミングで与えられたクロック切換用セクタ信号に応じた信号を生成し、この信号を前記クロック切換マルチプレクサへの前記切換信号として出力するクロック切換手段と、

前記クロック切換マルチプレクサより出力されるクロック出力にตอบสนองして動作するCPUと、

を備えたことを特徴とする低消費電力処理装置。

【請求項2】 請求項1記載の低消費電力処理装置において、前記クロック切換手段は、

基準クロックに同期したクロックによって前記セクタ信号を同期化する同期用フリップフロップと、

このフリップフロップによって同期化されたセクタ信号がアクティブになっておりかつ前記切換前クロックと切換後クロックが同じレベルに変化したタイミングでその状態がセットされ、前記セクタ信号が非アクティブになっており、かつ前記切換前クロックと切換後クロックが同レベルに変化したタイミングでその状態がリセットされる選択用フリップフロップとを備え、この選択用フリップフロップの出力を前記切換信号として送出することを特徴とする低消費電力処理装置。

【請求項3】 請求項1の低消費電力処理装置を第1処理装置として備えたとともに、

前記第1処理装置に含まれるCPUよりも低消費電力のCPUを含む第2処理装置と、

第1の処理装置及び第2の処理装置内の各CPUのレジスタファイル間の情報を同一に保つためのCPU間インターフェース手段と、

第1処理装置のCPUに接続する第1キャッシュメモリシステムと、

この第1キャッシュメモリシステムを制御する第1キャッシュメモリ制御手段と、

第1処理装置のCPUと第1キャッシュメモリシステムを介して入出力されるバス制御手段を含む第1バス手段と、

第2処理装置のCPUからのバス制御手段を含む第2バス手段と、

第1バス手段と第2バス手段を第1処理装置及び第2処理装置からの前記バス制御手段にตอบสนองして切り換え、外部のリソースに適合したタイミングで外部バス及び外部

2

制御信号を入出力するバススイッチインターフェース手段と、

第1処理装置の動作と第2処理装置のいずれか一方を選択して動作させ、選択されなかった処理装置内のCPUを停止させるか又はアイドル状態に保つモード切換手段と、

を具備したことを特徴とする低消費電力処理装置。

【請求項4】 請求項3記載の低消費電力処理装置において、前記モード切換手段は、前記第1及び第2処理装置に組み込まれており、第1又は第2処理装置の動作中の一方の処理装置から他方の処理装置へ動作を移行するときは、動作中の処理装置のモード切換手段は他方の処理装置の停止又はアイドル状態を解除して当該処理装置はアイドル状態に移行するように制御し、これによって起動した前記他方の処理装置のモード切換手段はそれまで動作中であった処理装置へのクロック供給を停止するように制御して当該他方の処理装置はそれまで動作中であった処理装置の状態を引き継いで動作するように制御することを特徴とする低消費電力処理装置。

【請求項5】 請求項3記載の低消費電力処理装置において、動作中の処理装置が自身のレジスタファイルを書き換えた時、その書き換え結果と一致するように停止中の処理装置のレジスタファイルも前記CPU間インターフェースを介して書き換えておくようにしたことを特徴とする低消費電力処理装置。

【請求項6】 請求項1又は2に記載の低消費電力処理装置において、前記CPUは、前記クロック切換用セクタ信号を発行してクロック周波数をダイナミックに切換えるクロック切換命令を具備したことを特徴とする低消費電力処理装置。

【請求項7】 請求項3ないし5の内の1つに記載の低消費電力処理装置において、前記第1及び第2処理装置のCPUは、前記クロック切換用セクタ信号を発行してクロック周波数をダイナミックに切換えるクロック切換命令と、前記モード切換手段により処理装置をダイナミックに切換えるCPU切換命令の少なくとも一方を具備したことを特徴とする低消費電力処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、省電力化を目指すCPU内蔵の処理装置に関するものである。

【0002】

【従来の技術】 近年、マイコン/パソコンやラップトップパソコン等のパーソナル計算機システムを中心としたコンピュータシステムの機能及び性能が向上すると共に、これらの処理装置が消費する電力が急速に高まって来ており、様々な面で問題となって来ている。一般の工業用やOA用、科学技術計算用、組込み用といった計算処理装置や制御処理装置も、自然環境へ与える悪影響への問題や、電力コスト、設環境コスト、電力供給事情等

3

の問題から、今後省電力への要求が高まって来ると予想できる。特にラップトップパソコン等のハンドヘルド

(持ち運び可能) 処理装置は、電力供給が電池であることが使用環境上多く、使用可能時間に直結する省電力化は必須である。

【0003】従来、省電力化を目指すパーソナルコンピュータ(PC)では、下記の2つの方法が最も省電力化に有効な手法として提案されて来た。その1つは、特開平8-241145号に開示されているように、低速クロックと高速クロックを用意し、アイドル状態(PCが有効な処理を実行していない状態)やバッククラウド処理(優先順位の低い処理)では、低速クロックをCPUに供給する様に切り換えることで動作電力を低減するものであり、他の1つは、特開平11-7344号に開示されているように、低速CPUチップ(一般に消費電力が小さい)と高速チップ(一般に消費電力が大きい)を用意し、2つのCPUからのバスを共通に接続し、低速処理でも問題の無い処理を実行する際には低速CPUチップ側に切り換えることで動作電力を低減するものである。またクロック速度を切り換える方法で、スムーズなクロック切換を行えるようにした技術が特開平10-91272号等に開示されている。

【0004】

【発明が解決しようとする課題】従来技術では以下の点で問題があり、その解決が課題となっている。

(a) 近年のCPUは内部動作周波数が非常に高くなってきており(300~600MHz)、CPUの外部(基板上等)でスムーズにクロックを切り換えることは実質不可能である。この点について、従来技術は何の解決策も開示していない。

(b) きめ細かな電力制御による低消費電力化が求められているが、従来技術はクロック切換やCPU切換のオーバーヘッドが大きく、その要求に応えることができない。

(c) CPUを切換える従来技術においては、一般のCPUチップを用いることが前提となっており、CPU間の内部情報の一致化処理のオーバーヘッドが大きくきめ細かな切換処理操作が困難である。

(d) 高速CPUにはキャッシュメモリや高速バスシステム等の高速化技術が用いられるが、低消費電力CPUを構築するに際して、その高速化技術の扱いや新たに必要となるアーキテクチャについて従来技術は何も開示していない。

【0005】本発明の目的は、クロック及びCPU切り換えをスムーズにかつ少ないオーバーヘッドできめ細かく行え、キャッシュメモリやバス切り換えも短時間で行えるようにして大幅な消費電力低減を可能とした処理装置を提供することにある。

【0006】

【課題を解決するための手段】本発明は、与えられた基

4

準クロックに同期しかつ互いに異なる周波数を持つ複数のクロックを生成するPLLクロックドライバと、このPLLクロックドライバにより生成されたクロックの内の1つを入力された切換信号に応じて選択し、切換後クロックとして出力するクロック切換マルチプレクサと、このクロック切換マルチプレクサにより切り換えの前に選択されていた切換前クロックと前記切換後クロックが同じ信号レベルにありかつ前記基準クロックに同期したタイミングで与えられたクロック切換用セレクト信号に応じた信号を生成し、この信号を前記クロック切換マルチプレクサへの前記切換信号として出力するクロック切換手段と、前記クロック切換マルチプレクサより出力されるクロック出力にตอบสนองして動作するCPUと、を備えたことを特徴とする低消費電力処理装置を開示する。

【0007】更に本発明は、上記の低消費電力処理装置において、前記クロック切換手段は、基準クロックに同期したクロックによって前記セレクト信号を同期化する同期用フリップフロップと、このフリップフロップによって同期化されたセレクト信号がアクティブになっておりかつ前記切換前クロックと切換後クロックが同じレベルに変化したタイミングでその状態がセットされ、前記セレクト信号が非アクティブになっており、かつ前記切換前クロックと切換後クロックが同レベルに変化したタイミングでその状態がリセットされる選択用フリップフロップとを備え、この選択用フリップフロップの出力を前記切換信号として送出することを特徴とする低消費電力処理装置を開示する。

【0008】更に本発明は、低消費電力処理装置を第1処理装置として備えたとともに、前記第1処理装置に含まれるCPUよりも低消費電力のCPUを含む第2処理装置と、第1の処理装置及び第2の処理装置内の各CPUのレジスタファイル間の情報を同一に保つためのCPU間インターフェース手段と、第1処理装置のCPUに接続する第1キャッシュメモリシステムと、この第1キャッシュメモリシステムを制御する第1キャッシュメモリ制御手段と、第1処理装置のCPUと第1キャッシュメモリシステムを介して入出力されるバス制御手段を含む第1バス手段と、第2処理装置のCPUからのバス制御手段を含む第2バス手段と、第1バス手段と第2バス手段を第1処理装置及び第2処理装置からの前記バス制御手段にตอบสนองして切り換え、外部のリソースに適合したタイミングで外部バス及び外部制御信号を入出力するバススイッチインターフェース手段と、第1処理装置の動作と第2処理装置のいずれか一方を選択して動作させ、選択されなかった処理装置内のCPUを停止させるか又はアイドル状態に保つモード切換手段と、を具備したことを特徴とする低消費電力処理装置を開示する。

【0009】更に本発明は、上記の低消費電力処理装置において、前記モード切換手段は、前記第1及び第2処理装置に組み込まれており、第1又は第2処理装置の動

作中の一方の処理装置から他方の処理装置へ動作を移行するときは、動作中の処理装置のモード切換手段は他方の処理装置の停止又はアイドル状態を解除して当該処理装置はアイドル状態に移行するように制御し、これによって起動した前記他方の処理装置のモード切換手段はそれまで動作中であった処理装置へのクロック供給を停止するように制御して当該他方の処理装置はそれまで動作中であった処理装置の状態を引き継いで動作するように制御することを特徴とする低消費電力処理装置を開示する。

【0010】更に本発明は、上記の低消費電力処理装置において、動作中の処理装置が自身のレジスタファイルを書き換えた時、その書き換え結果と一致するように停止中の処理装置のレジスタファイルも前記CPU間インターフェースを介して書き換えておくようにしたことを特徴する低消費電力処理装置を開示する。

【0011】更に本発明は、上記の低消費電力処理装置において、前記CPUは、前記クロック切換用セレクト信号を発行してクロック周波数をダイナミックに切換えるクロック切換命令と、前記モード切換手段により処理装置をダイナミックに切換えるCPU切換命令の少なくとも一方を具備したことを特徴とする低消費電力処理装置を開示する。

【0012】

【発明の実施の形態】以下、本発明の実施の形態を詳細に説明する。まず、計算機の消費電力を考察すると、消費電力は主としてCPUの処理能力に比例している。すなわち下記の様な事情から計算機システムのCPUの消費電力は上昇してきており、今後も上昇していくものと考えられる。

【0013】(1) 年々大きくなりつつあるOS等の大規模な機能プログラムをストレスなく実行する処理能力がCPUに要求されており、そのための機能ハードウェアやCPU処理スピードを直接的に高めるためのハードウェア(パイプラインやキャッシュメモリ及びそれらの制御ハードウェア等)の量がそれに伴って大きくなってきている。これは単位時間当りのトランジスタのスイッチング量が増加することを意味しており、特にCMOSプロセス主体の近年のCPUはトランジスタのスイッチング量に消費電力が比例することから、消費電力の増大を招いている。

【0014】(2) CPUを高速化するための主力技術として、CMOSプロセスを微細化(近年では0.1~0.2 μ mプロセスが用いられつつある)して、高集積化とトランジスタのスイッチング速度向上を実現し、そのプロセスの性能向上を利用して、CPU(又はCPUコア)へ供給するクロックの周波数を向上させていく手法が待たれている。近年では、CPUチップの内部では500MHz級のクロックが用いられることも珍しくない。プロセスの高集積化特性は、ハードウェアの並列化

等に用いられ、スイッチング速度向上との相乗効果によりパイプライン等のレイテンシ短縮、すなわちクロック周波数の向上に寄与することになる。しかし、(1)で述べたトランジスタのスイッチング量はその分相乗的に増加することとなり、大幅な消費電力上昇につながっている。

【0015】(3) CPUを設計する際の低消費電力化の手法としては、CPUが動作する必要のない時のクロック停止等によって電力を制御する低消費電力モードの具備、低電圧動作化(3.3V、2.2V等)が現状用いられている。しかし、前者は、通常運転時には低消費電力化されていない為、有効な処理時間に対する省電力化ではない。また、後者は、近年低消費電力化に大きく寄与してきた手法であるが、あまり大きく電圧を下げすぎるとトランジスタのスイッチング速度又は駆動能力自体を低下させCPUの動作スピード等の性能を悪化させ兼ねないし、信号のノイズマージンの点でも好ましくないと考えられる。

【0016】上記(1)~(3)のCPU設計上の低消費電力化、省電力化に関する考慮から、(3)のCPUの低電圧動作を除けば、CPUが有効な処理を実行している時に、いかに単位時間内のCPU内の有効スイッチングゲート数(又はトランジスタ数)を減らせるかが、CMOSプロセス高性能CPUの低消費電力化のポイントになることがわかる。これには、回路論理規模の小型化によって有効動作トランジスタ数そのものを減少させる手法と、クロック周波数を低下させる手法とをうまく組み合わせていくしかない事も(1)、(2)より明らかである。しかし、(1)(2)でも述べた様にこれらの手法は、CPUの高性能化と完全に逆行するものであり、その矛盾を回避するために、あまり高速処理が要求されていない処理や、実質アイドル処理と見なせる処理に対して効果的にこれらの手法を適用可能な、CPUのアーキテクチャや省電力の為の命令セットを提供していく必要がある。すなわち、動作中にいつでもどんなタスクや処理ステップに対しても、自在にかつ、高応答にて必要な様々な電力消費モードに移行でき、その電力制御動作をプログラム中の命令や、外部からの指令で実行できるきめ細かな低消費電力化制御が自在にかつ動的に可能なCPUを提供する必要がある。例えば、高消費電力だが高速化に最適化されたCPUと、低速だが低消費電力化に最適化されたCPUとを1チップにまとめ、自在に切り換えて動作させる様にし、かつ、各CPUに与える基本クロック周波数も自在に切り換えることを可能にする電力制御用命令セットを有し、プログラム中に必要な動作スピードに応じて細かく電力制御用命令を配置する様にすれば良い。

【0017】図1は、上記の方針に基づいて構成した、本発明の低消費電力処理装置の構成例を示すブロック図で、高消費電力だが高速に動作する高速CPUコア20

と、それほど高速ではないが低消費電力の低速CPUコア30と、両CPUコアに対してそれぞれ周波数可変のCPUクロック11、81を供給するクロック周波数切
換回路10、70とが設けられている。但し低速CPU
コアに対するクロック81は周波数一定でもよく、その
ときはクロック周波数の切換回路70は不要である。C
PUコア20と30の間は各コア内のCPU間インター
フェース回路21、31を介して接続され、2つのCP
Uコア内のレジスタファイル22、32間のコヒーレン
シ制御（内容を同一に保つ制御）や、CPUコア間のス
テータス情報のやりとりに用いられる。各CPUコアに
は、キャッシュメモリ40、50がそれぞれ装備されて
いることもある。特に高速CPUコア20には処理効率
を向上させるために外部I/Oやメモリシステム等の外
部リソース90上の情報をCPU内部に蓄えておくため
の高速アクセス可能なキャッシュメモリシステム40が
必須であるケースが多い。外部リソース90（メモリ、
I/O、システムバス等）へのアクセスを実行するため
に、各CPUコアからのバス手段B1、B2がそれぞれ
存在するが、2つのCPUコアのうちアクティブな方の
みを外部リソースに接続するためにバススイッチイン
ターフェース60を具備している。バス手段B1、B2は
キャッシュメモリシステム40、50が存在する場合に
は一度CPUコアからのキャッシュメモリ制御信号C
1、C2と共にキャッシュメモリシステム40、50に
それぞれ接続され、キャッシュメモリ上に情報が存在し
ない場合のみバススイッチインターフェース60を介し
て外部リソース90から情報を入出力する。なお、両C
PUコアがキャッシュメモリを持つ時は、キャッシュメ
モリ間のコヒーレンシ制御を行うためのキャッシュ間イ
ンターフェース41、51を設ける。

【0018】以上に示した図1の構成で、クロック周波
数切換回路10、70を各CPUコア20、30に対し
て設けているが、これらは、各CPUコアへのクロック
周波数をその時の状況（処理内容や環境等）に応じて自
在に切り換え、最適な処理速度や消費電力となるように
各CPUコアの動作を制御するための手段を提供する。
これと、高速CPUコア20、低速CPUコア30の切
り換えによって、全体としてより最適な処理速度及び消
費電力となるようにに処理装置全体を動的に制御する事
で、従来より効率的な電力管理/動作速度管理を動作中
にダイナミックに実施することが可能となり、結果的に
システム全体の総合的な消費電力を減少させることがで
きる。

【0019】クロック周波数切換回路10は、動作の基
準となる基本クロックCLから複数の互いに厳密に同期
したクロックを生成するPLL（フェイズ・ロックド・
ループ）クロックドライバ12と、そこからの複数のク
ロック17から1つを選択するためのクロック切換マル
チプレクサ14と、PLLクロックドライバからの切換

制御同期クロック16（クロック17のうちのどれかを
使用しても良い）に同期してセレクト信号SELAに対
応した切り換え信号15を生成してマルチプレクサ14
へ送出するクロック切換制御回路13とから成る。マル
チプレクサ14は、切り換え信号15によって選択され
たクロック11をCPUコア20に与え、CPUコア2
0はこの動作用クロック11に基づいて動作する。クロ
ック周波数切換回路70も同様な構成であるが、出力す
るクロック81の周波数は異なっている。

【0020】上記のクロック周波数切換回路10の構成
において、PLLを内蔵したクロックドライバ12を用
いている。これは、従来の様に単なる分周回路によっ
て複数のクロックを生成する場合には、a) 外部入力に
最も高速なクロックを用意する必要がある、b) 生成で
きる周波数の自由度が低い（1、1/2、1/4、1/
8…倍となる）、c) 基本クロックに対し出力クロック
の遅延が発生してしまう等の問題があるが、PLLを使
用すればこれらすべての問題を解決できるからである。
すなわち、基本クロックCLは低周波数のものであつて
も、内部のVCOで非常に高い周波数のクロックを生成
し、これを分周して基本クロックとの位相比較を行うよ
うにすればよく、かつVCOからの内部クロックを使用
して複数種のクロックを生成するように構成すること
で、生成できる周波数の自由度も大きくできる。またV
COを複数種持つこともでき、生成できる周波数の自由
度はさらに大きくなる。さらに、マルチプレクサ等の遅
延量に相当するディレイ要素を介してPLLに基準とな
るフィードバッククロックを戻すことにより、目的とす
るクロック出力の遅延を基本入力クロックに対する出力
クロックの遅延を除去したり、調整したりするのが容易
になる。

【0021】図2は、クロック切換制御回路13の構成
例を示すもので、この回路ではPLLクロックドライバ
12からのクロック17が2つのクロック（高速クロッ
クと低速クロック）の場合である。クロック選択信号S
CINは、図1のセレクト信号SELA又はSELB自
体か、あるいはそれに応答して生成した信号であり、そ
れをフリップフロップ（FF）107、106にて同期
用クロック16に同期した信号、即ち、PLLクロック
ドライバ12からの複数のクロック17と厳密に同期化
した信号とする。この信号はフリップフロップ106の
Q出力及びQN出力の互いに反転した2つの出力として
とり出され、3入力NANDゲート102、101へ入
力される。NANDゲート102、101ではこれらと
高速及び低速クロックとのNAND論理がとられ、それ
らNANDゲート出力は、ゲート104、103で構成
されたラッチ用RSフリップフロップ113のセット端
子S及びリセット端子Rにそれぞれ入力される。このR
Sフリップフロップ113のセット側の出力であるゲー
ト104の出力がクロック切り換え信号15としてクロ

ック切換マルチプレクサ14へ出力される。

【0022】図3は、図2に示したクロック切換制御回路13の動作を示すタイミングチャートを示すもので、高速クロックは低速クロックの2倍の周波数としている。これら高速及び低速クロックと同期用クロック16はすべてPLLクロックドライバ12から同期して出力されている。クロック選択信号SCINがハイレベルに遷移し(時刻t0)、それをフリップフロップ107、106により同期化して得られたフリップフロップ106のQ出力がハイレベルに変化した後、最初に高速クロックと低速クロックがともにハイレベルに遷移するタイミング(時刻t1)を使ってクロック切換信号15がローレベルからハイレベルに切り換わり、クロック出力11も高速クロックから低速クロックに切り換えられる。逆に、クロック選択信号SCINがハイレベルからローレベルに遷移し(時刻t2)、それをフリップフロップ107、106により同期化して得られたフリップフロップ106のQN出力がハイレベルに変化した後、最初に高速クロックと低速クロックがともにハイレベルに遷移するタイミング(時刻t3)を使って、クロック切換信号15がハイレベルからローレベルに切り換えられる。それに応じてクロック出力11も低速クロックから高速クロックへ切り換えられる。ただし、時刻t3ではすでに高速、低速クロック共ハイレベルの状態になっているので、クロック切換信号15の変化タイミングで直ちに切り換えが行われる。ここで切り換えタイミングt1、t3は、同期用クロック16や高速、低速クロックとの間にゲート通過時間等から成る適当なディレイtc_{sd1}、tc_{sd2}を有しており、確実にその変化点は高速、低速クロックレベルが共に安定してハイレベルにある時に存在する設計となっている。これによって、切り換え時のハザード等の発生が防止される。

【0023】なお、図2に示したように、同期用クロック16の反転クロック108又はクロック16をディレイゲート109で遅延させたもの、あるいは、タイミングさえ満たせば(マルチプレクサ14のA1、A2入力の状態が安定しているタイミングで信号15が変化するのであれば)、同期用クロック16そのものを用いてラッチ用RSフリップフロップ出力をフリップフロップ105でシフトして、上記のタイミングを満足するように調整したものをクロック切換信号15として用いても、同様にハザード等を防止できる。

【0024】図4は、図2に示したクロック切換制御回路の変形例で、高速クロックと低速クロックの排他的論理和を(EXOR)ゲート114でとり、その反転出力を2入力のNANDゲート101'、102'に供給するようにしたもので、他は図2と同じである。この構成によると、高速クロック、低速クロックが共にローレベルの時でもハイレベルの時でも、単に同一レベルでさえあればクロック選択信号の変化点を基準に切り換え動作

を実行する回路を実現することが出来る。なお、EXORゲート114からのハザードを除去したい場合は、ゲート114の出力を同期用クロックの正転又は反転、又は適当にクロックをディレイさせたクロックで駆動されるフリップフロップ115を介してNANDゲート101'、102'に供給すれば良い。なお、EXORゲート114からのハザードにより、クロック切換信号15にハザードが乗ったとしても、マルチプレクサ14がトランスファergeートスイッチを用いて構成されているので、後述するCMOSトランスファergeートの特性によって、入力(A1、A2)の状態が同一レベルに安定しているタイミングで切換信号が変化する限り(ハザードも含む)、マルチプレクサ14の出力にハザードが発生することはないと考えてよい。従って、フリップフロップ115は存在した方がよいが、必須の要素ではないと考えられる。

【0025】図5は、図2の高速・低速の2クロックを切り換えるマルチプレクサ14の構成例を示しており、CMOSプロセスで構成したトランスファergeートスイッチである。トランジスタ110P、111PはPタイプトランジスタと呼ばれ、そのゲート(S入力が接続された端子)に負電位が印加されていて、A1(高速クロック)入力にこれに対して高い電位が印加されると導通し、A1入力の電位をバッファ112Bへ伝える。一方、トランジスタ110N、111NはNタイプトランジスタと呼ばれ、ゲートに正電位が印加されているときA2入力にこれより低い電位が印加されると導通し、A2(低速クロック)入力の電位をバッファ112Bへ伝える。従ってS入力の電位によってA1、A2入力の一方をバッファ112Bを介して端子Zへ出力でき、1~2段のトランジスタでバスが構成されているから高速スイッチが可能である。そして、上記の条件以外の状態では各トランジスタの出力側はハイインピーダンスとなるが、バッファ112Bの入力インピーダンスもCMOS回路故にハイインピーダンスであるので、トランジスタ出力とバッファ入力との間の接続部には電荷が蓄積され、しばらくの間はその状態になる直前の状態が維持され、これは出力端Zにも反映される。従って、クロック切換信号15(S入力)の切換時に、インバータ112Gの遅延によって2つのトランジスタゲート入力がともに低レベルとなる瞬間があっても、A1入力、A2入力に印加される電位がそのとき安定していれば、バッファ112Bからハザードが発生することなく、スムーズなクロック切り換えが行われる。

【0026】図6は、2つのクロックを切り換えるマルチプレクサの別の構成例で、クロック切換制御回路の一部機能も有したものである。この回路の基本スイッチ600は、図5と同様にCMOSプロセスで構成したトランスファergeートスイッチであるが、CPUからのセレクト信号及びスイッチ600の切換信号15の形式が異

11

なっている。即ち、基本スイッチ600は、切換信号SA1が“0”のときA1入力、切換信号SA2が“0”のときA2入力を出力する。入力されるセレクト信号SelA1、SelA2はそれぞれそれが“1”のときクロックのA1入力、A2入力を選択するためのCPUからの信号で、EXORゲート602とNANDゲート603、604によって、SelA1=“1”、SelA2=“0”のとき信号SA1=“0”、SA2=“1”となって入力A1が出力され、SelA1=“0”、SelA2=“1”のとき信号SA1=“1”、SA2=“0”となって入力A2が出力される。さらにこれらゲートの働きで、SelA1、SelA2がともに“0”かともに“1”のときは信号SA1=SA2=“1”となり、基本スイッチ600が完全にオフし、A1、A2両入力が出力されるのを防いでいる。

【0027】また、レジスタ601を用いてラッチクロックLCLによりいったんセクタ信号をラッチして基本スイッチ600へ送っているが、これは切換タイミングをクロックと同期化すると、ラッチして同時に切換信号SA1、SA2を出力することで、切り換えのタイミング誤差を極小化し、基本スイッチ600の2入力がショートしないようにする機能を持つ。ここで前者の同期化であるが、このためにはラッチクロックLCLを基本クロックCLに厳密に同期化させておく必要がある。この方法としては、CPUからの切換指令を基本クロックCLに同期させて出力するのが一般的である。

【0028】図8は、図6の回路を多段結合して4入力クロックCL1~CL4を切り換えるようにした回路例で、図6の基本スイッチ600を多段結合し、それらを、セレクト信号CL1~CL4から図6と同様にレジスタ804にラッチクロックLCLでラッチして生成した切換信号で切り換えるようにしたものである。なお、各基本スイッチ600出力端Zに接続されるバッファ(図6の112B)は各スイッチには不要で、多段結合最後の出力端にバッファ803を設けるだけでよく、これによってクロックドライバとする。

【0029】以上では高速・低速の2クロックを切り換えるためのクロック切換制御回路及びマルチプレクサの構成例と動作を図2~図6を用いて説明したが、より多くのクロックを切り換えるための各回路例を次に述べる。図7(a)は、4つのクロックCL1~CL4の1つを選択してとり出す回路の例で、図1のクロック切換制御回路13とマルチプレクサ14に相当する回路である。マルチプレクサ701~703は図5に示した2入力切換用のものであり、これらの多段結合でスイッチ部分が構成されている。切換信号はレジスタ704の出力で与えられ、その切換タイミングは、図6で説明したのと同様の、基本クロックに同期したラッチクロックLCLで与えられる。レジスタ704へ入力される選択信号

12

SelE1、SelE2は、CPUからのセレクト信号SelCL1~SelCL4を図7(b)のようにエンコードして得られた信号である。これら選択信号SelE1、SelE2、レジスタ704、ラッチクロックがクロック切換制御回路の機能を実現していると考えられるが、選択信号とラッチクロックを生成する部分は図7では省略されている。なお、ここでは4クロックの例を示したが、さらに多数の切り換えも同様に構成できる。

【0030】図7は、図5のマルチプレクサを多段結合した構成例であるが、図2または図4で説明したクロック切換制御回路13とマルチプレクサ14を組み合わせた回路を1つの単位回路として、これらを多段結合しても多数のクロック切り換えが行われる。この場合、切り換える前と切り換えた後のそれぞれのクロックのみに依存して、それらのクロックが同レベルの論理にある時に切り換えるタイミングをセットする方法によって同様にスムーズな切り換え制御を実現する事ができる。すなわち、初段の回路において関係の無いクロック入力をマスクした後、AND論理をとってNANDゲート101、102に入力するか、あるいはEXOR+反転論理(EXNOR)をとってNANDゲート101、102に入力すれば良い。例えば一方のクロック入力を“1”に固定すれば(マスク)、AND論理をとった場合でもEXOR+反転論理をとった場合でも、他方のクロックを通過させる単なるバッファとなり、そのクロックが次に切り換わるべきクロックの場合にのみそのクロックと同期化されたクロック選択信号とで切り換え動作が行われる事になる。また、両方のクロック共関係なければ、その回路ではクロック選択信号自体が変化しないはずであり、マルチプレクサ14の出力も“1”に固定されるはずである。この2つの条件から、最終的に生きているクロックバス、すなわち、現在選択されているクロックバスと、次に選択されるべきクロックのバスとを切り換えているマルチプレクサ及びその切り換え制御回路1組のみがこれまで説明してきた基本となる2入力クロック切換制御回路として動作しているとみなせる為、同様のスムーズな切り換え動作が可能となる訳である。なお、新しく切り換わるべきクロックバス選択はCPUがフェッチした切換命令をデコードして対応する各マルチプレクサへのセクタ信号を特定し、それらを必要な論理に設定すれば良い。

【0031】図9は、多段入力時のクロック周波数切換回路の例とその中でPLLクロックドライバの使用例を示す。切換回路911~917の各々は、例えば図5に示したマルチプレクサに図7のようにレジスタをつけ加え、このレジスタに周波数fのクロックを用いてセレクト信号をラッチして同期化させるようにしたもので、これは図7の構成と同様である。このように、PLLからの最高周波数のクロックで同期化すれば、厳密な基本クロックへの同期化が可能になる。あるいは図6の回路

をこれら切換回路としてもよい。

【0032】図9では、フィードバックすべきクロック周波数を f として、それを基準に $f/32$ の周波数の基本クロックを外部から入力し、 $f/2$ 、 $f/3$ 、 $f/4$ 、 $f/6$ 、 $f/10$ 、 $f/16$ 、 $f/22$ 、 $f/32$ 及び f の9種の周波数を生成可能なPLLクロックドライバを想定した。なお、周波数 f のクロックは、同期用クロックとしても使用している最も高速なクロックである。PLLクロックドライバ901は、基準クロックCL($f/32$)をリファレンスクロック(REFCLK)として受け、フィードバッククロックCLBを内部で $1/32$ 倍した信号の位相とリファレンスクロックの位相とをほぼ一致させる様にして上記各9種のクロック位相を調整している。すなわち生成される各クロックは、最も高速な周波数 f のクロックに厳密に同期して生成されるため、フィードバッククロックCLBの遅延量に応じてその分だけ早い位相にてPLLクロックドライバから出力されていることになる。従って、切換回路911~914の列、同915、916の列、及び同917の列の各遅延量に相当する遅延補正回路921~923を图示した様に同期クロック(f)へ挿入して各段への同期用クロックを構成すると共に、遅延補正回路923出力をフィードバッククロックCLBとしてPLLクロックドライバに戻せば、CPUへ出力する最終段のクロック出力CLoutの位相を結果的に基本クロック位相とほぼ一致させる厳密な同期化が可能となる。なお、PLLクロックドライバから生成されるクロック出力の位相は、ループフィルタ(抵抗、コンデンサで補正)を外付け可能なPLLを用いれば微調整が可能である。

【0033】クロック周波数切換のためのPLLの使用は、今後CPUが高速化されるに従ってCPUチップ内に内蔵されることが必須となってくると考えられる。すなわち、現状でもLSIの内部クロックで500MHz級のCPUが存在し、外部のシステム(電子ボード上)で直接扱える周波数ではない。従って、外部の基本クロックとしては100MHz以下程度に抑え、LSIチップ内部で高速周波数動作をさせる構成が必須となってくる。さらに図1の様なCPUシステムは、1チップLSIとしてまとめないと成立しなくなってくる事を意味しているとも言える。すなわち、500MHzを越えるクロックを厳密に同期化させて切り換え制御を行うためには、PLLクロックドライバを含むクロック周波数切換回路10、70とCPUコア20、30は少なくとも一体化して1チップ化する必要性が生じ、理想的には1チップCPUとして図1のシステム全体をを集積していくことが自然な流れである。現状はこの様な思想の低消費電力1チップLSIは存在しないが、1チップCPUに集積していくことを考慮した場合の構成上の特徴、新たに実現できる機能及び効果等について図1をベースに以下に検討を加える。

【0034】クロック周波数切換回路とCPUコアとを集積化する利点及び必要性は上述したとおりである。従って2つのCPUコア20、30を1チップLSIに集積化することによる機能及び効果をまず検討する。第1に2つのCPUコアを1チップ化することにより、CPU間I.F.回路による通信レイテンシの短縮、スループットの飛躍的な向上が図れる。例えばレジスタファイル22、32間のコヒーレンシ制御をリアルタイムで実現できる。動作中のCPUが自身のレジスタファイルを変更した時、書き込み対象レジスタ情報と変更データ情報及びライトコマンドを並行して他方のCPU内のレジスタファイルに送り、対応するレジスタ内容を実時間で変更しておくことが可能となる。これにより、CPU間の切り換えが発生しても改めてコンテキストのコピーを実施する必要はなく、余分なオーバーヘッド無しに直ちに処理を開始できるという効果が得られる。

【0035】第2に、クロック周波数切換回路10、70への切り換え信号や前述した入力クロックのマスク信号等から成るセレクト信号SELA、SELB及びCPU間の切り換え制御信号/情報等を低レイテンシで高速生成できる様になる。従来では、これらの信号を生成する手段は、CPUの外部にステータスレジスタ等として用意し、CPUのMOV命令やロード/ストア命令等を用いて値をセット、リセットする方法を採らざるを得なかったが、1チップ化することによりCPUの命令として、組み込みプログラム中に適切に配置することによって、それをCPUがフェッチした時、直ちに実行できる様に構成することができる。CPUコア内では結果的にクロック周波数切換回路10、70内の非同期の同期化処理用FF107、106を除去できるか、又はFF106のみで同期化可能とできる場合もあり、またCPU内部動作周波数は外部に対してはるかに高速であるという利点もある為、切り換えにかかるレイテンシを大幅に短縮することができる。CPU間の切り換え用制御信号の結線例やその切り換えプロトコル例は後述するが、これらのCPU間情報伝達がすばやく行え、それに対応する動作や処理も高応答ですばやく実行できるという利点が生まれる。結果的に、プログラム中できめ細かな電力制御を記述でき、さらなる低消費電力化を図ることができる効果が得られる。

【0036】次にキャッシュメモリ部(キャッシュメモリシステム40、50)の1チップLSI内蔵化を考えてみる。まず第1に、レジスタファイルのコヒーレンシ制御と同様に、1チップ化によってマシンサイクルに同期した高速なキャッシュ間インターフェースが構成できるため、リアルタイムでキャッシュメモリの内容を同一に保つことが可能となる。具体的には、動作中のCPUのキャッシュメモリが書き換わった時、並列して他方のCPUのキャッシュメモリの同じアドレス部分を書き換える処理を行う。動作中のCPU側から送る情報は、目

的とするキャッシュメモリアドレス及びキャッシュデータと書き込みコマンド信号であり、書き込みコマンドがアクティブになったら対応するキャッシュメモリのアドレスの内容を目的とするキャッシュデータにリアルタイムで書き換えていく。従来では、CPUチップが別々であることから、CPUが動作を再開した時には一度前のキャッシュ情報をフラッシュ（無効にする）する必要があったが、本方法によりCPU切り換え時のキャッシュメモリに関するオーバーヘッドを極少化することができる。なお、キャッシュメモリシステムを各CPUコアそれぞれに用意する場合は、メモリシステムの構造を高速対応タイプと低消費電力対応タイプに分けてそれぞれ設計、前者を高速CPUコアに、後者を低速CPUコアに割り付けると、最も消費電力、動作速度の点で総合的に最適な構造を提供できる。また、キャッシュメモリ部のみを1チップ化して、別に1チップ化されたCPU部とバス手段B1、B2、キャッシュメモリ制御信号C1、C2等の信号群を外周で結合する方法も考えられるが、すべてを1チップ化する場合に比べて、明らかにCPUとリソース間のアクセスレイテンシは増大し、CPUシステムの性能にはマイナスであると言える。

【0037】第2に、1チップ化により、2組のキャッシュメモリを1組にする事も可能である。具体的には、1つのキャッシュメモリを2つのCPUコアが共有し、より有効にキャッシュメモリを利用する為に、高速CPU側が動作中はクロック同期型のアクセス制御等で高速アクセスを実行し、低速CPU側が動作中は、リード/ライトコマンドによるスタティックなアクセス制御（CMOSプロセスの場合、コマンドによりアクセスしている時以外は消費電力がほぼゼロ）により低消費電力化を目指した方式を採れば良い。これによれば、キャッシュメモリ間のコヒーレンシ管理/制御が必要なくなるだけでなく、多量のトランジスタを必要とするメモリ部を1組分除去できることで、他の1組のキャッシュメモリ容量を増やしたり、チップのサイズを小さくすることができたり等、さらなる高速化又は低消費電力化を目指すことができる。この場合も必然的に上述した様なコヒーレンシ制御や共有キャッシュメモリを実現するために、キャッシュメモリの制御アルゴリズム及び、キャッシュメモリシステムの構造は両CPU共通であるという前提で、CPU側のキャッシュアクセス制御回路が設計されなければならない。この場合、バススイッチインターフェース60は、2つのCPUからのバスを切り換える方式のものではなく、一体化されたキャッシュメモリシステムへのアクセスが失敗した時に、直ちに外部へのアクセス動作が発生する様に制御される入出力用のバススイッチバッファとして構成しても良い。

【0038】バススイッチインターフェースユニット60は、別のLSIにまとめ、リソース90の近くに配置する方式も有効である。特にキャッシュメモリシステム

を1つにまとめて2つのCPUで共有化する場合、バス手段B1、B2はキャッシュメモリで結合され、キャッシュメモリからはそのキャッシュメモリへのアクセスに失敗した時（キャッシュメモリ上に必要とするデータが存在しない時）に、外部リソースへのアクセスに向かう場合に必要となるバス手段が1つあれば良いことになる。この内部バス手段をCPUシステムのLSIチップの外部アクセスバスとして定義すれば、バススイッチインターフェースユニット60は、リソースやシステムバス等へのインターフェースとなるバスインターフェースユニットとして定義することもできる。これをリソースへのアクセス用制御信号やデコード、入出力バッファ等と一緒にLSIに集積すれば、各リソース又はローカルリソースブロックの近くに、1つ又は分散して複数配置する方法を採ることができる。集中して集積されたCPUチップから配線を引きまわす場合と比較して、バッファ遅延やデコード遅延等が除去できる（リソースのそばに必要となるバッファICやデコードの代りにバスインターフェースユニットを代用できる）分、高速化が図れる可能性がある。

【0039】次に図10、図11により、CPU間の切り換え制御について説明する。まず図10に、CPU間の切り換え制御及びクロック周波数制御のための信号接続を示す。高速CPU20、低速CPU30はそれぞれクロックセレクト制御回路202、203を有している（CPU内部に内蔵していても良い）。CPUはクロック周波数設定/変更命令やCPU切換命令をフェッチし実行を開始すると、対応するクロックセレクト制御回路202又は203が必要な制御信号（Select、Stop等）を生成してクロック選択ドライバ200、201を制御する。ここで、図10に示したクロック選択ドライバ200、201は、図1のクロック周波数切換回路10、70の機能に相当する。また、信号Select（A）、（B）は、前述したクロック切換マルチプレクサの切換信号やクロック入力マスク信号等のコントロール信号を含むところの図1におけるセレクト信号SELA、SELBの機能に相当し、信号Stop（A）、（B）は、図1のクロック停止信号CLSTA、CLSTBの機能に相当するものとする。

【0040】図10では、動作中に選択されているCPUのみにクロックが供給されている構成を想定して、互いに相手側のCPUによって自身に供給されるクロックの起動/停止処理が実行される様に、Stop信号をクロスに結線している。図11は、高速CPU20から低速CPU30への切り換えが実施される様子を図示しており、これについて詳しく説明する。まず動作中のCPUである高速CPU20がスイッチ命令（CPU30へのスイッチ）をフェッチすると、CPU30への再起動指令を生成し、アイドルステートへ移行する。再起動指令は、クロックセレクト制御回路202へのクロック制

17

御指令23を伴い、CPU30へのクロック再起動を実行する(stop(B)を解除)と共に、CPU間通信ライン61を介してCPU30にクロック再起動後直ちにアイドル状態から抜け、次の処理ステップに移行する様に指令する。CPU30側はクロックが再起動されると、まずCPU20側のクロックを停止させるべく、クロックセレクト制御回路203を介して、stop(A)をアクティブにした後アイドル状態を抜け、次の処理ステップに移行する、なお、CPU30は、自身のレジスタにコピーされたCPU20の情報を10 用い、CPU20に引き続いて必要な処理を実行して行く為、処理手順や処理内容に矛盾を生じることは無い。逆方向の、CPU30からCPU20への切り換えも同様である。

【0041】以上の様に、CPU間のスイッチ命令とCPUへのクロック周波数切り換え命令とをリアルタイム電力制御命令としてCPUの命令セットに組み込み、プログラム中でそれらの命令を自由に使用できる様構成することができる。それらリアルタイム電力制御命令の使用法について以下に説明する。

【0042】シーケンサ等の制御用コントローラにリアルタイム電力制御命令を使用する場合、図12に示した様に、各タスクのサンプリングタイムの違いが優先順位の違いに直接関係することが多い。すなわち、サンプリングタイム(図中三角印で指示)の間隔が短いタスク程、高応答な処理を要求される傾向が高く、優先順位が高いタスクであると言える。従って、これらのタスク間でタスクスイッチ(図中のa~g)が発生する時、起動されたタスクの先頭又は、タスクスイッチをサポートするOSの中で電力制御命令を実行し、結果的にその優先30 順位に従って適切なCPU及びクロック周波数を選択する。例えば、最も優先順位の高いタスクTA1は高速CPUコア20を選択してクロック周波数も最高のものを選択する様にし、2番目の優先順位であるタスクTA2は低速CPUコア30のクロック周波数最高の選択設定に、最下位の優先順位であるタスク3は低速CPUコア30のクロック周波数中程度の選択設定という様に、処理タスクに応じてCPUコアとクロック周波数を選択する。この様にタスクの優先順位が下がるに従って消費電力を少なくする選択モードに移行すると効果的である。40 また、どのタスクも実行しないスリープモード又はウェイトモード(サスペンドモード)等のアイドルモード時は、低速CPUコア30のクロック周波数最低の選択設定にしておけば、無駄な消費電力を極力抑えることができ、さらに効果的である。

【0043】図13は、マルチプロセッサ動作における各CPUのタスク処理の一例であり、プロセッサ間同期機構によって関連のあるタスク間で同期をとり、スケジュールされた並列処理を進めている様子を示している。SYN C m - n はCPU m とCPU n が同期する事を示50

18

しており、idleは空き時間(アイドル時間)を、NOPは何も処理しない状態(遊びタスク)を示している。本図で、idleとNOPはタスクをプリスケジュールする際に認知できるので、そこに電力制御命令を配置し、適切な省電力モードに設定(例えば低速CPUコアのクロック周波数最低に設定)すれば効果的に省電力化を図ることができる。

【0044】次に、バッテリーシステムを用いて動作中の処理装置、例えばノートパソコンにおけるパワーコントロール手段及び利用技術の一例を挙げる。

(1) 動的に消費電力を調整するパワーコントロール手段と、バッテリーの残量を検出するバッテリーモニタ手段を有するバッテリー駆動型処理装置において、プロセスシステムをバッテリーで稼働させたい目標可動時間を設定した時、バッテリーの残量を参照しながら動的に消費電力を調整することにより、目標可動時間を保証する。

(2) 上記の(1)において、処理装置が処理するアプリケーションに対して処理負荷に応じて優先順位をつけ、より負荷の軽い処理ほど消費電力を下げ(=処理速度も下げ)て実行することより、高負荷アプリケーション実行中にオペレータに負荷を感じさせずに、消費電力を調整する。

(3) (1)又は(2)において、スーパーバイザモードとして処理装置のユーザーが直接処理の優先順位を選択できる手段(外部ハードスイッチ又はソフトウェアスイッチ等)を設け、選択した優先順位に対応した消費電力に調整(パワーコントロール)する。この操作は動作中にリアルタイムで実施しても良い。このような(1)~(3)の利用法で用いるパワーコントロール手段は、前述したCPU切換え(高速/低速CPU切換え)とクロック周波数切換えとのコンビネーションによる電力制御手段を用いると最も効果的である。

【0045】次に、同様に本発明のパワーコントロール手段を用い、処理装置(PC等)のCRT上のウィンドウ画面にて消費電力設定を細かく行う方式での電力制御(パワーセーブ)について一例を述べる。

(1) マルチタスクアプリケーションの優先度設定を行うコントロールボタンを各ウィンドウに設けるか、または各アプリケーション起動時に優先度の設定ウィンドウを表示し、マウスにて優先度の設定を行う。消費電力設定コントローラは、各アプリケーションの処理頻度のコントロールや、全アプリケーションの優先状態、PCの全体消費電力及び電源状態から最適なクロック周波数を決定し、これによりユーザーの期待するPCのパフォーマンスを効率よくCPUのパワーに割り当てることができ、かつ低消費電力化が図れる。

(2) PCの端末から離れる場合、消費電力設定ウィンドウ画面にて、離れる時間、バックグラウンド高速演算が必要なアプリケーションの指定を行う。消費電力設定コントローラは、ディスプレイなど必要の無いデバイス

の休止を即座に行い、利用者が戻るまでに行う演算量と消費電力、電源状態の関係から最適なクロック周波数を決定する。これによりユーザがアクセスしない間の期待する処理を効率良く実行しかつ低消費電力化を図れる。ここで端末から離れた事をセンスする仕掛けとしては、入力デバイス（キーボード、マウス等）による検出、あるいは設定ウィンドウ画面において直接低消費電力モードへの移行の設定を行う等の方法を用いる。

（3）使用しないデバイスの指定をウィンドウにて指定し、休止状態とする。これによりユーザの利用状況に応じて消費電力を低減できる。

【0046】ここで、消費電力設定コントローラとは、やはり前述した本発明の電力制御手段（CPU切換えとクロック周波数切換えのコンビネーション）を操作する前述の電力制御命令セットを用いて記述されたプログラムモジュールである。なお高速コントロールが必要な場合、このコントローラはファームウェア化されていても良い。CPU以外のファームウェア実行システムが存在する場合、前記電力制御命令セットは、外部からのビット線又は情報入出力手段等によってその実行システムから直接命令実行の指示ができ、それに応答してCPU、システム内の対応する電力制御命令を直ちに実行させる外部命令指示手段を設けておく必要がある。ファームウェア実行システムは、この外部命令指示手段を用いて必要な電力制御のための手続きを実行する。

【0047】以上、利用技術について述べて来たが、これらの手続きを記述するプログラムは、プリコンパイラで従来のプログラムに電力制御命令を条件にしたがって挿入して自動生成し、システムにダウンロードしたり、OSのドライバやシステムコールとして組み込み、OSが必要に応じて起動する仕掛けを用意したりする事でユーザーに直接負担をかけずに実現することができる。

【0048】また、図1に示した処理装置は、最も実現し易い形態として2つのCPUを含む形を提示しているが、3つ以上のCPUを含む処理装置を構成することも可能である。そのうち有効に動作しているのは1つのCPUだけであり、そのCPUが動作中に変更したレジスタの内容は、2つのCPUで構成する場合と同様、選択されていない他のCPUのレジスタに実時間でコピーし、コピーレンシを保つことができる様にCPU間インターフェースは構成され動作する。CPU間の切換え時は、切換え命令中で切換わるべきCPUを指定することになる。命令が実行されると、指定されたCPUと現在選択されているCPUとの間で本実施例中で説明した2つのCPU間の切換え操作手続きが実施される。

【0049】この様な複数のCPUを含む処理装置は、フォールトトレラントシステムや異常時の代行処理付高信頼性システム等に応用することも可能である。すなわち、実行中の命令処理に異常を感知した場合（例えばバリエーション等）、CPU間の切換え手段を用いて他の

CPUに切換え、命令を再処理し、誤動作を回避する等の高信頼化動作が可能となる。なおこの場合、エラーが発生した命令によるレジスタの変更を実施しないか、又はいくつかの過去の処理ステップのレジスタ内容をCPU内に保持しておき、再実行すべき時点の状態に選択されたCPUのレジスタ内容を復帰させる手段を備える必要がある。

【0050】

【発明の効果】本発明によれば、以下の効果が得られる。

（1）動的にかつきめ細かく高速に電力を制御することができ、それによって処理装置の低消費電力化を図る効果が得られる。

（2）低価格な1チップCPU化が可能な技術を提示しており、低消費電力化が必要な様々なアプリケーションに高いコストパフォーマンスを確保しながら適用できる効果が得られる。

【図面の簡単な説明】

【図1】本発明になる低消費電力処理装置の構成例を示すブロック図である。

【図2】クロック切換制御回路の構成例である。

【図3】図2のクロック切換制御回路の動作を示すタイミングチャートである。

【図4】図2の回路の変形例である。

【図5】マルチプレクサの構成例である。

【図6】クロック切換回路の例である。

【図7】クロック切換回路の別の例である。

【図8】多数クロックの切換回路の例である。

【図9】多数のクロック周波数切換回路の例である。

【図10】CPU切換制御の結線例である。

【図11】図10の動作説明図である。

【図12】シーケンサシステムなどのタスクスイッチング、サンプリングタイム及び優先順位の例を示す図である。

【図13】マルチプロセッサ動作に於ける各CPUのタスク処理例を示す図である。

【符号の説明】

- 10、70 クロック周波数切換回路
- 12 PLLクロックドライバ
- 13 クロック切換制御回路
- 14 クロック切換マルチプレクサ
- 20 高速CPUコア
- 21、31 CPU間I.F.回路
- 22、32 レジスタファイル
- 23 バス制御手段
- 30 低速CPUコア
- 40、50 キャッシュメモリシステム
- 41、51 キャッシュ間I.F.
- 60 バススイッチインターフェース
- 61 CPU間通信ライン

21

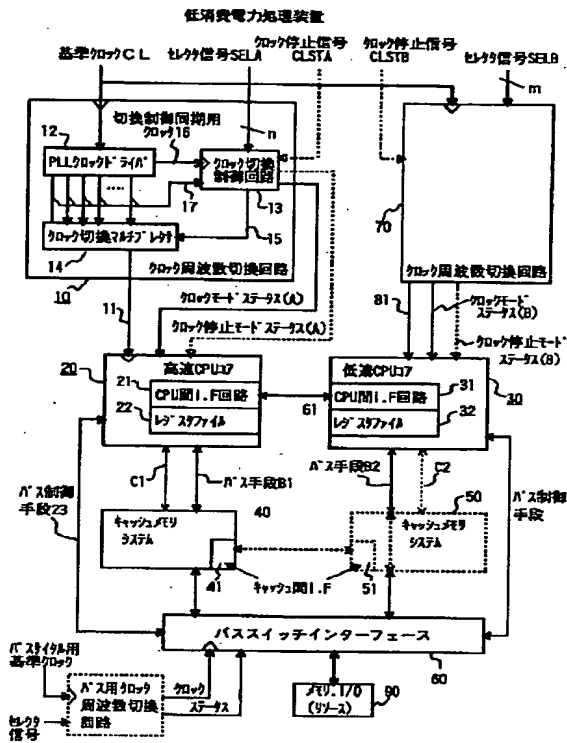
90 外部リソース

101、102 多入力NANDゲート

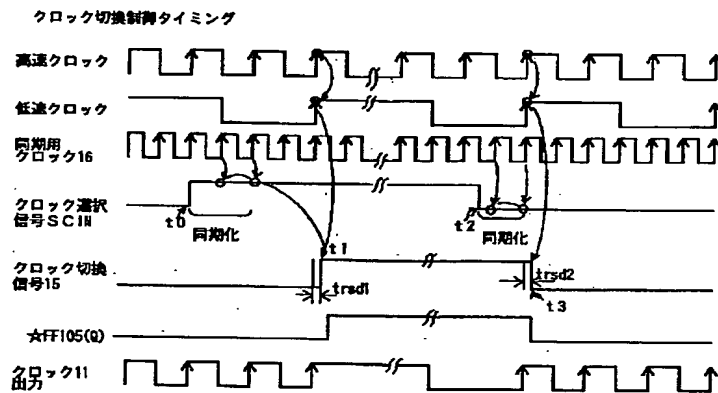
106、107 同期化処理用フリップフロップ

*

【図1】



【図3】

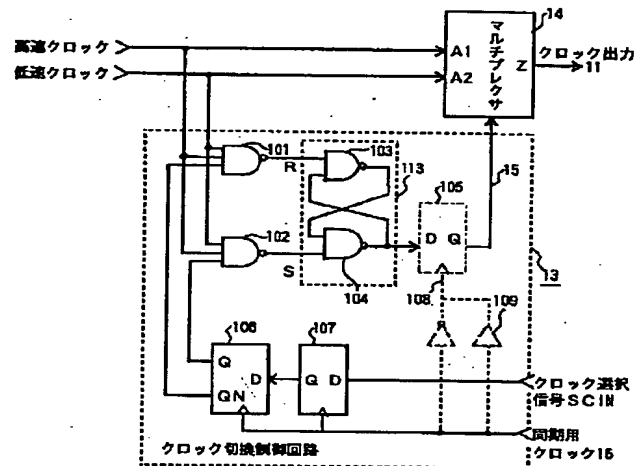


22

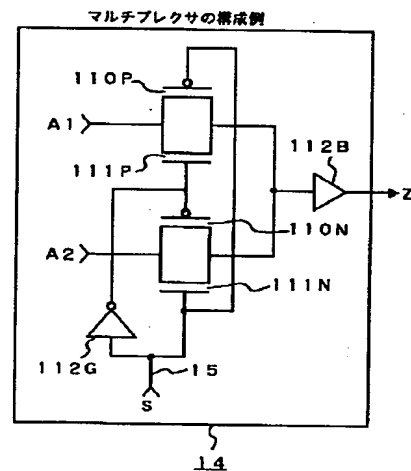
*113 ラッチ用RSフリップフロップ

114 EXORゲート

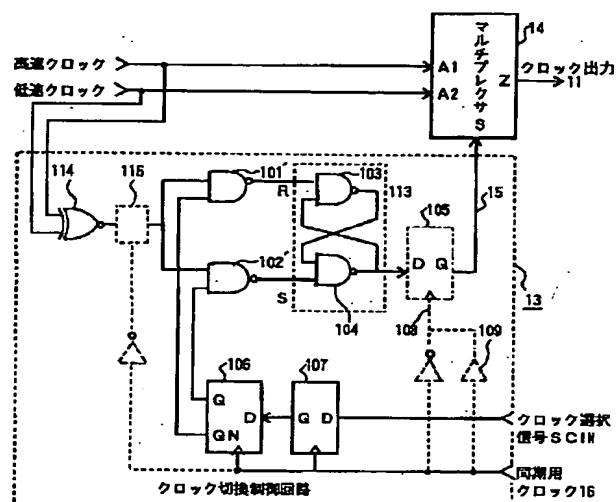
【図2】



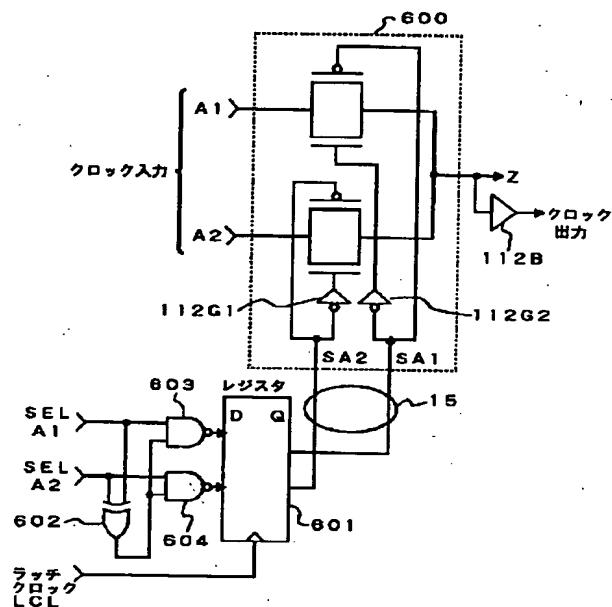
【図5】



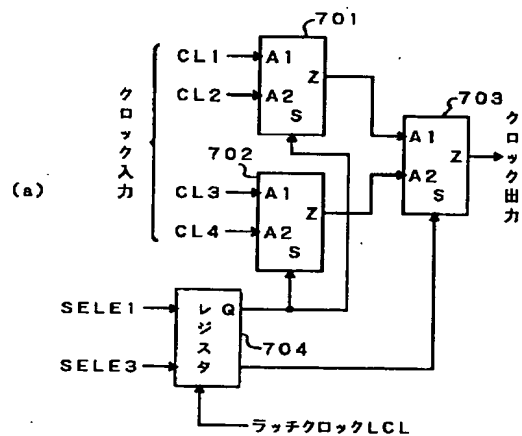
【図4】



【図6】



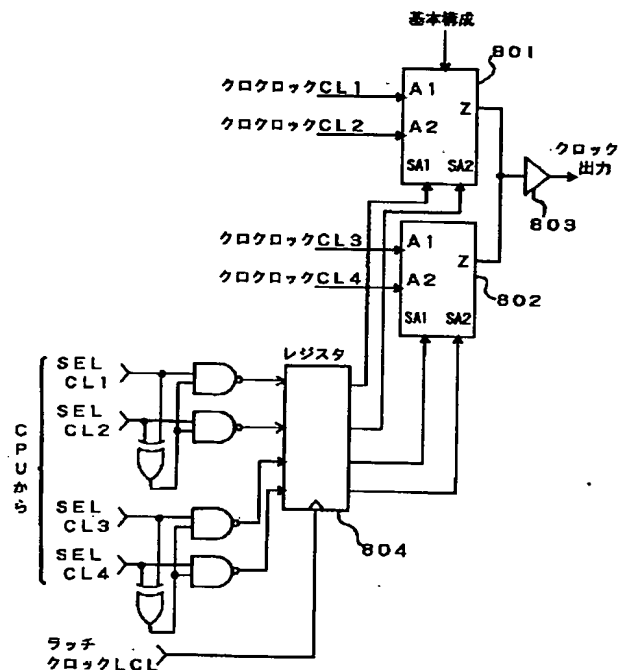
【図7】



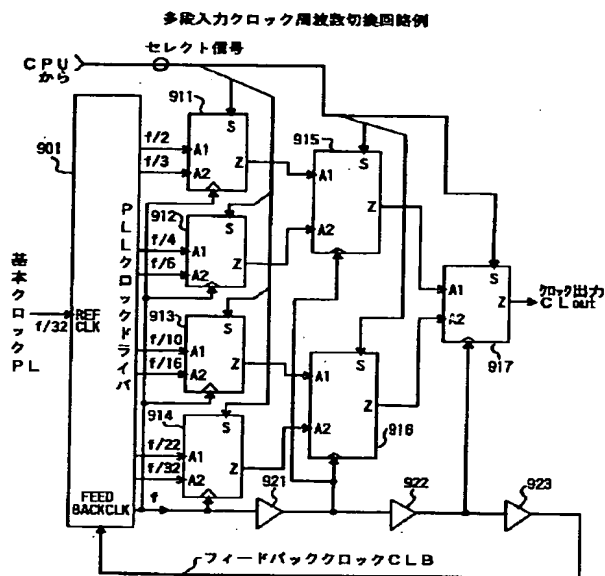
(b)

SELCL1	1	0	0	0
SELCL2	0	1	0	0
SELCL3	0	0	1	0
SELCL4	0	0	0	1
SELE1	0	1	0	1
SELE3	0	0	1	1

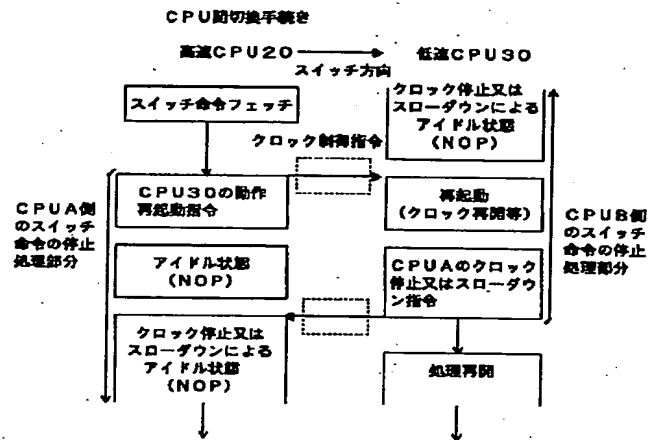
【図8】



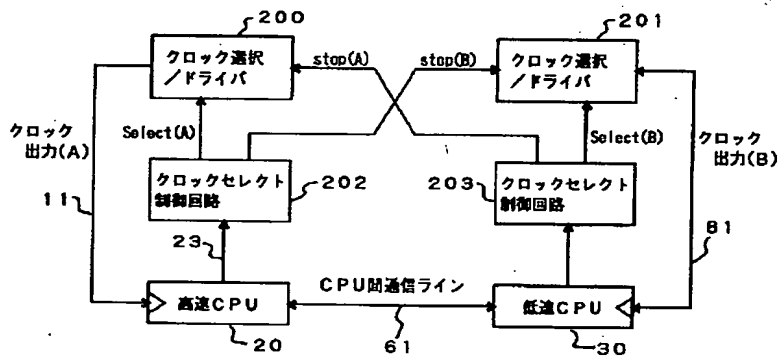
【図9】



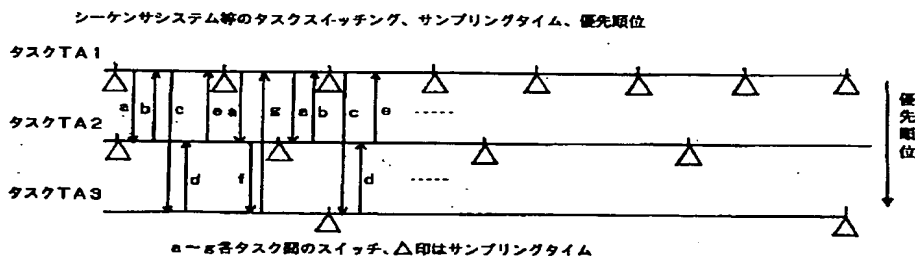
【図11】



【図10】



【図12】



BEST AVAILABLE COPY

【図13】

マルチプロセッサ動作における各CPUのタスク処理例

CPU0	CPU1	CPU2	CPU3	CPU4
タスク0-0	タスク1-0	NOP	タスク3-0	タスク4-0
タスク0-1	idle			idle
SYNCO-1	タスク1-1	タスク2-0	SYNC	3-4
タスク0-1	SYNC	1-2		タスク4-1
idle	タスク1-2	タスク2-1	NOP	
	SYNCO-	1-2-3-4		

フロントページの続き

(72)発明者 山本 健次郎
 茨城県土浦市神立町502番地 株式会社日
 立製作所機械研究所内

(72)発明者 小山 昌宏
 茨城県土浦市神立町502番地 株式会社日
 立製作所機械研究所内

(72)発明者 桃井 康行
 茨城県土浦市神立町502番地 株式会社日
 立製作所機械研究所内

(72)発明者 船津 輝宣
 茨城県土浦市神立町502番地 株式会社日
 立製作所機械研究所内

(72)発明者 池 勝久
 茨城県土浦市神立町502番地 株式会社日
 立製作所機械研究所内

Fターム(参考) 5B079 AA06 BA02 BB01 BC01